ESL-IC-05-10-40

EnviroDB : APPLIED DATABASE SYSTEMS DESIGN FOR THE NATIONAL ENVIRONMENT ASSESSMENT TOOLKIT (NEAT)

Sean Hay Kim PhD Student
Viraj Srivastava PhD Student
Azizan Aziz Researcher
Carnegie Mellon University
Pittsburgh, Pennsylvania

ABSTRACT

The General Service Administration (GSA) "Workplace 20 20" project aims to investigate the relationship of physical environment, building attributes, and "best practice workplace strategies" to workers' performance and organizational effectiveness [1]. In order to conduct the study, the Center for Building Performance and Diagnostics at Carnegie Mellon University (CBPD) uses a suite of Post Occupancy Evaluation (POE) tools, NEAT, which consists of existing and newly developed tools and techniques.

EnviroDB, one of the NEAT tools, integrates all the data and procedures used in the project. It is designed to help the user analyze multi-factorial causality intuitively as well as meet the fundamental functional requirements of generic database systems. A relational data model (RDM) is proposed for EnviroDB.

The RDM would be a proper strategy for EnviroDB. As the information treated in the project is aggregated and representing of a group of building components. The data granularity of the project would be an entity of RDM. Second, the manipulation of multiple conditioned queries in RDM is relatively handy while maintaining the data normality. Third, the process of data model design/deployment is efficient and easy to get supported compared to other data model designs.

INTRODUCTION

Post Occupancy Evaluation (POE) and building diagnostics are critically important in determining the overall building performance with respect to environmental quality, such as thermal, IAQ (Indoor Air Quality), acoustic, visual, spatial and building integrity [1]. The POE typically examines building legacy information, indoor environmental measurement and testing, occupant survey and interviewing processes. Due to such different processes and criteria, POE adopts a combination of tools, techniques and devices in data gathering, interpretation and analysis.

The "Workplace 20 20"/NEAT project sponsored by GSA has applied the methodology of POE. Beyond assessment and evaluation of building functions , the goal of the project is to investigate the relationship of physical environment, building attributes and "best practice workplace strategies" to workers' performance and organizational effectiveness [1]. Add-on methods identifying such relationship have been devised and integrated into the typical POE technique.

NEAT TOOLS

The Center for Building Performance and Diagnostics at Carnegie Mellon University (CBPD) has developed and is continuously refining a unique and proprietary method for NEAT. The method consists of five tools; Environmental Instrumentation (EnviroBot), Occupant Satisfaction /Collaboration Survey (EnviroQuest), Technical Attributes of Building Systems Survey (TABS), Real-time Physical Indicator Identification Software (EnviroSoft) and Central Database Systems (EnviroDB).

The current generation of EnviroBot includes sensors measuring air temperature at 3 heights (TEMP), radiant surface temperature (SUR_TEMP), relative humidity (RH), carbon dioxide (CO_2), carbon monoxide (CO), volatile organic compounds (VOC), particulates (PM), air velocity (AIR_VEL), light levels at 3 locations (LGH), and a photometric camera that analyzes brightness/contrast and glare. The sensor outputs are recorded using GUI based data acquisition software that runs on a laptop computer on the cart. Another toolkit measures acoustical quality [2]. TEMPs, RH, CO_2, CO, VOC and PM are also measured on 3 independent continuous measurement stands.

EnviroSoft enables the user to record the physical indicators (such as fans, warmers and etc.) in a tablet PC, during an expert walkthrough, and perform the analysis and obtain and visualize the results instantaneously on-site [1]. EnviroSoft uses a unique form of input based on selecting physical indicators that are then automatically allocated to the desired space [1]. EnviroSoft imports measurement from the EnviroBot sensors and data loggers in order to complete the data analysis[1]. Physical indicators and spaces can be hyperlinked to pictures and documents to store additional information [1].

EnviroQuest includes two types of occupants' response; User Satisfaction Questionnaire and Collaboration Questionnaire. The User satisfaction questionnaire captures occupants' satisfaction about physical environment of individual workspace and work area. The purpose of collaboration questionnaire is to assess office workers' collaboration experience with regard to office spatial settings [3]. Both the questionnaires use an intensity gradient indicator and text field as their answer format. They are distributed to the occupants during the measurement of the environmental parameters.. The skeleton of the questionnaires is basically maintained over the whole project. Depending on characteristics of the site and organization, some of questions are customized for a deeper understanding.

EnviroDB gathers all the primitive data from each tool and sorts them into the designated category. EnviroDB enables the user to create, retrieve and modify data and to watch the data transaction. In addition, by different query options, it provides the user filtered results based on what the user wants to see and inference results that allow the user to view the information trend. With the proposed web deployment, the user will not need to manage the data manually and the visualized summary will be reported to the user.

DESIRED SPECIFICATION OF EnviroDB

Functionalities of database are based on online transaction processing (OLTP) and decision support systems (DSS). These two features of generic database are realized in EnviroDB.

Online Transaction Processing(OLTP)

Data storage

As a term of data acquisition gets longer, the increased amount and complexity of data gets harder to be handled in a plain spreadsheet. If the data acquisition is not long-term, but is bulky, a simple retrieval of desired data might not be done easily.

A single measurement of a GSA site results in a great deal of data. Before the measurement, building information such as floor plans are gathered. Preliminary occupancy questionnaires are launched to give the evaluator a feeling of which aspect should be considered first. During the measurement, environmental data from sensors, questionnaires, pictures and expert's suggestions are collected. Continuous measurements of environmental data contribute to the size of measurement data extensively since continuous data are recorded every minute for at least 24 hours. After the measurement, analysis of brightness/contrast and glare and further recommendations are added.

Typically around ten sets of measurement data are added every year. As the same instruments are used and due to the data transfer between analyzers, primitive data and analyzed data could be mixed up unless there is an integrated system of data storage. Post-measurement processes worsen the inherent difficulty.

Various types of input data

The types of inserted data are in a variety of data formats. For instance, environmental measurement data are numeric whereas questionnaire data are text as well as numeric. Images such as pictures and drawings extend the variety of the data formats.

The format of questionnaire answers are not uniform even if they have a numeric form. Some questions require a single-choice while others are multiple-choice. For cases where there is no appropriate answer for a question, comment fields are provided with almost every question. Therefore the database should accommodate diverse data type.

Flexible data model

There is always a possibility that a format of a tool could change. For example, questionnaires could be customized by the site and the organization. A measurement criterion could be added or modified if a new sensor is attached . Since the format of the basic data input could change, the database interface and queries also need to be changed accordingly unless the data model has the required flexibility. Therefore the structure of the data model should be designed to catch data type change pliantly rather than fixed data field.

Automation of data acquisition

Since input data are from multiple sources and most of input data are batch-processed (i.e. instead of a single data insertion, a group of data is poured into the data storage), the data acquisition should be automated . Therefore a conversion and mapping logic between primitive data and processed data should be well-defined. And an interface between each tool and the database should be stream-lined such that the human effort to manage data acquisition is minimum.

Decision Support System (DSS)

Usually the POE process is time-consuming and labor intensive when the data analysis is done manually. Consequently, a single factorial analysis (i.e. how is the occupants' satisfaction level as the light level gets higher) is likely to be prevalent. However, occupants' sensation and productivity would not be proportionally controlled by a single factor unless the factor has a direct sensitivity on human behavior.

There are lots of potential combined factors with different conditions in the project. i.e. how is the occupant satisfaction level in a workspace with the VAV system but without a perimeter unit? How are the other cases with perimeter units?. Countless combinations of factors could be made to identify the sensitivity of the factor and relationship between a group of factors and the user satisfaction/work productivity.

A properly-designed DSS is to help analyzers compile useful information from the primitive data to identify the relationship between building characteristics and occupant/organization behavior. DSS features such as user-conditioned query, muti-factorial analysis, graphical data trend/tendency, comparative study and summary reporting will be included in EnviroDB.

User-conditioned query

A list of query templates could be formulated from basic queries which are commonly asked during result analysis. For example, descending sort from spaces with highest air temperature, or portion of unsatisfied occupants with air temperature in a workgroup despite thei air temperature measurement being within comfort thermal range. The questions list up and the user drills down for further queries.

When the analyzer attempts to examine another point of view on the same data set or when the predefined query is not adequate for the building/organization configuration, the base queries need to be customized by changing conditions. The database should allow the user to specify query conditions to modify/add the query template.

Easy multi-factorial analysis (multiple search option)

Another objective of NEAT is to verify the assumptions when the questionnaire was prepared. For instance, an occupant with a VAV system and an individual controller would be more satisfied with the thermal condition than another occupant without them.

However, there are many factors, some of which are seemed to be relevant, some of which might not have such a strong causality. To verify the causality between some factors, a set of potentially relevant factors has to be fixed and permutations of those factors have to be tested.

Permutations might not be easily tested unless there is a supporting automation such as a database system. The biggest advantage of applying database in the data analysis process would be the ease of combining multiple conditions. As long as the database adopts an easy and flexible data model in terms of development of multi-factorial analysis, verifying the causality by testing permutations would be faster.

Graphical trend/tendency

There are as many results as the number of permutation cases in the multi-factorial analysis. When those results are shown numeric, it might not be intuitively and easily understood. Representation of data should have a form of graph or matrix, where factors are described in an axis and results are shown in another axis. Intensity of level has to be distinct using color differentiation.

Because the measurement criteria are physically originated from building geometry, gradation of measured values flowing along building floor plan would be extremely helpful to identify building characteristics. This need precipitates incorporating the data model into Geographic Information System (GIS), which a proposed future task.

Comparative study

With a graphical representation of data trend/tendency, accompanying comparative cases besides the base case would be as important to identify an effect of the factor on building performance and human reaction. There would be binary conditions (with/without) and gradient conditions (i.e. with 20% contribution of T-80 lighting fixture).

Comparative study could be a break-through over all DSS features of database when it is realized in conjunction with graphics.

Summary reporting

Statistics and summary have been classic requirements of DSS. Getting statistic reports takes time because that is concentrated information after going through all base data. Therefore rather than applying dynamic conditions, to offer summary templates to the user and to let them select one is a more popular way of implementation.

SYSTEM ARCHITECTURE

The three tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three layer architectures a popular choice for Internet applications and net-centric information systems [4].

The user system interface top layer includes user services such as data input and display. The middle layer is in charge of process management. Transactions such as information delivery and resource allocation/sharing between the user interface layer and

the database layer are controlled by the process manager. The database management layer stores physical data, retrieve/modify data and format of data upon requests of the process manager.

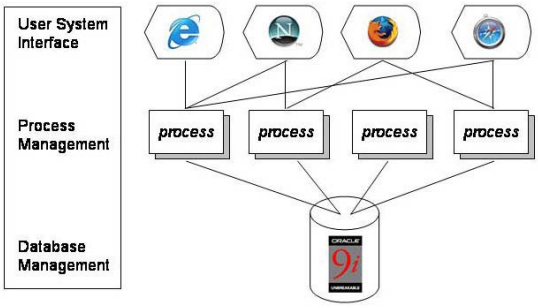

Figure 1. Three tier distributed client/server architecture

The user reaches EnviroDB systems using web browsers such as MS explorer and Apple FireFox. Apache HTTP server deploys as Web Application Server (WAS). Java Virtual Machine (JVM) on top of Apache manages data transactions and asynchronous queuing. Oracle 9i is chosen for the Database management systems (DBMS).

SOFTWARE ARCHITECTURE

Model-View-Controller (MVC) is a software architecture that separates an application's data model, user interface, and control logic into three distinct components so that modifications to the view component can be made with minimal impact to the data model component [5].

The model is the domain-specific representation of the information [5]. Applications are operated on the model. The applications retrieve data and let the view reflect the change upon data change. The controller takes an event such as user actions or batch process triggers and drives a change of views and model. The view is usually the user interface and presents the model or part of it.

In EnviroDB, JAVA is choosen as a developing language since it is platform-independent and useful extensions such as XML parser are easily obtained for free. The view of MVC is developed by Java Server Page (JSP). The controller is implemented with Java Servlet and the model is done with Java Data Object.

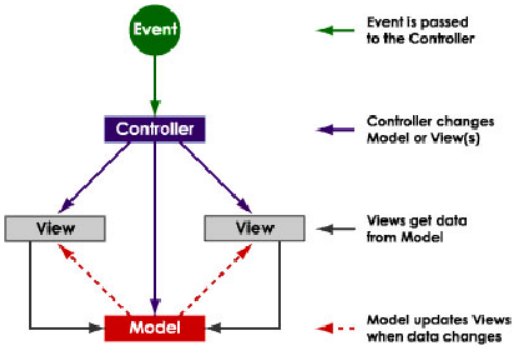

Figure 2. Model – View – Controller (MVC) architecture [5]

PROPOSED DATA MODEL

Relational data model (RDM) is proposed for EnviroDB since the RDB design would suit the above desired performance and specifications by following reasons.

First, the information treated within the project is aggregated and representing of a group of building components. (data granularity)

There are four actors governing the data flow in the project. They are building, work group, work space and occupant. Practically an occupant takes surveys per each work space. So that building, work group and work space are hinged points from which all information transaction starts.

The information retrieval would not reach down to single individual building component level. For example, type of ceiling light lamps in a work group might be asked and portions of each type might be asked if there are different types. Questions in the surveys ask the occupant mostly about their sensation and feeling about either building level, work group level or work space level. The finest element of data model does not necessarily need to be a building component such as individual lighting fixture or window.

If the data model is based on an object-oriented concept, the information is fragmented to each component level (i.e. after lighting fixture type is to be assigned to every individual lighting fixture object, the portion of specific lighting fixture in a work group is retrieved from dividing the number of the specific lighting fixtures by the number of all lighting fixtures) By this approach, both inserting data into each element and getting a statistical result such as “40% of lighting fixture in workgroup 5 is T-12.” might not be convenient and fast enough.

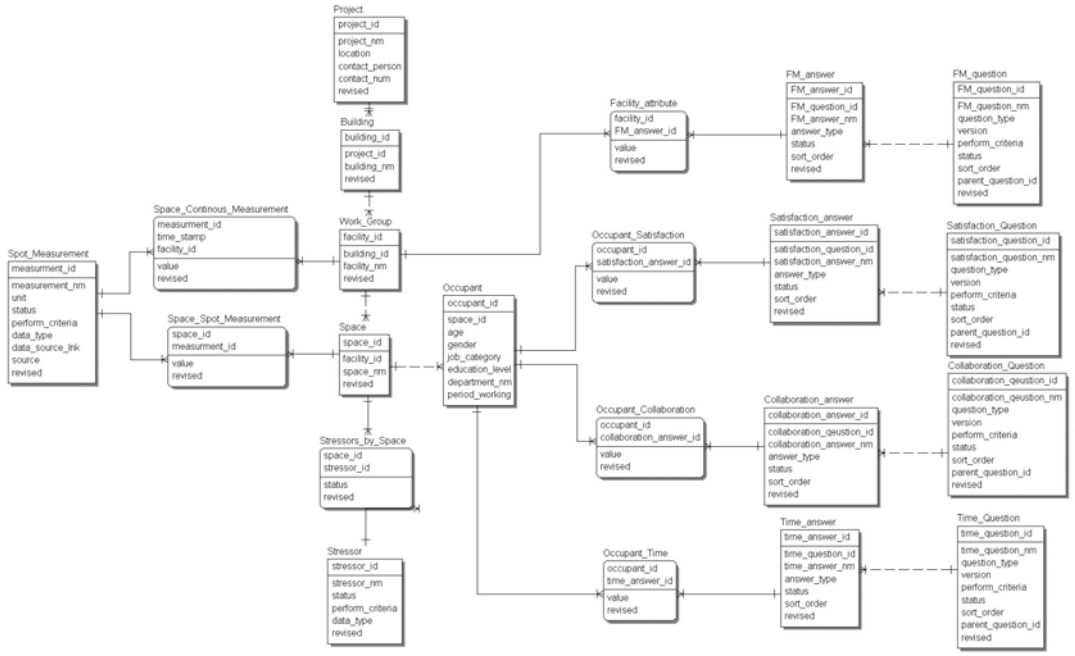

Figure 3. Entity Relationship Diagram (ERD)

Entity of RDM would be a reasonable resolution for the project. An entity is a “thing” or “object” in the real world that is distinguishable from all other objects [6]. It does not necessarily need to match as exactly as the real component. As long as it has a set of properties of which data are abstract enough to be informative in the project and is distinguished uniquely by identification, the entity is sufficient to be the finest set of data for the project. For example, satisfaction questionnaire and collaboration questionnaire are both questionnaires. But they have different structure and different questions to each other. Separating each questionnaire as an entity would be a reasonable design rather than letting them inherit from the questionnaire class.

Second, the manipulation of multiple conditioned queries is relatively handy while maintaining the data normality.

The most convenient reason to utilize a relational model is that it is relatively easy to define queries and to apply multi-factorial queries compared to other designs (such as hierarchical models or object-oriented models) while keeping the data set normalized.

The initial phase of designing RDB starts from clarification and identification of business logic. Each entity is identified by primary key and the relationship between entities is specified by foreign keys and the cardinality. So that a properly-designed RDB is optimized to the business logic and secures the data integrity. For instance, during the data normalization attributes of unnecessary entity could be dissolved into child entity or abstracted to parent entity.

A critical point of “join” queries (multiple conditioned queries) is to identify each key of a table. If the data integrity is not secured, results on the identical data sets might not be the same. If too many keys are needed to be traced and the hierarchy among keys is obscure, a query plan might not be sufficiently clear as well as the optimum query design is overwhelmed by the structural complexity.

Whereas, in RDB, the identification of keys from each entity is very explicit and tracing entities along the relationships is manifest. As long as the query designer has a thorough knowledge of the database schema, he/she is able to get a simple, but determinate query plan.

Third, the process of data model design/deployment is efficient and easy to get supported.

Process of RDB design is abbreviated to characterization of data sets. In other words, it is to determine a sufficiently fine data granularity (entities) and to associate an entity with other entities.

Once the dependency between entities is settled down and dominant entities are confirmed, RDB design gives the designer a pretty much of flexibility. For

example, a referenced relationship could be made by non-identifying foreign key or by identifying foreign key (primary key). It depends on the designer considering an ease of deployment, data migration, interoperability with other Database Management Systems (DBMS) and so forth.

As long as the mainframe is fixed, the design alternatives establish the model flexibility. Not only with a minor design change, the RDB design is able to cope with a design scheme change provided that the data flow is sustained. This can be done very quickly and does not require tremendous human and cost resources that would usually happen during system turnover or tuning.

As the logical data model and physical data model are almost identical, if there is, with a little variation, the deployment of RDB is comparatively straight-forward and fast. By taking advantage of a CASE tool such as Erwin or Rational Rose, a script even including system level schema such as procedure or trigger is automated. The CASE tool helps database administrator (DBA) manage the database easily by providing useful functions.

Besides, well-described design/tuning guide of SQL is plentiful and examples/best practices are very popular over almost every design case because the relational data model is much favored in terms of practicality in industrial applications.

Entity Relationship Diagram (ERD)

The point that all actors (Building, Work Group, Workspace, and Occupant), questions/answers, stressors, measurement criteria are treated as independent data aggregation is emphasized. The relation between entities is described in a mapping table such as Stressors by Space, Occupant_Satisfaction. i.e. While satisfaction question and answer tables contain the format of questions and answers, Occupant_Satisfaction table contains actual value of questions/answers from occupant. Stressor table, measurement tables have the same structure and relationships are defined in the corresponding mapping table. This data model schema is able to catch the input data format changes and useful when in need of tracking data deviation.

MODELING / DEPLOY TIPS

Numberless applications have adopted the relational database design such that useful guides/tips of implementation are literally abundant. There are some hands-on modeling/deployment tips for EnviroDB.

Utilize domain dictionary

The domain dictionary defines type of common data field. The domain dictionary is a data field library specifying physical and logical attribute of data. When all data fields come from the domain dictionary, naming convention and their physical data type are consistent over the data model. (i.e. a data field ‘name’ is defined as VARCHAR2(30). ‘measurement_name’ and ‘sterssor_name’ inherit from ‘name’ with the identical VARCHAR2(30). In the case ‘name’ is changed to VARCHAR2(40), ‘measurement_name’ and ‘sterssor_name’ are also changed to VARCHAR2(40) automatically).

Take a full advantage of Code table

When a status (such as ‘active’ or ‘obsolete’) or a category (such as ‘thermal’ or ‘IAQ’) of a datum is to be identified, instead of hard-coded values that only data modeler knows, assigning code value (such as that ‘active’ corresponds ‘01’) is recommended. This is easy to be managed and consistent in the whole data model. Particularly, utilizing code is very beneficial when values of code get changed because only the value field of code needs to change. Moreover, when another DBA takes over the database, it reduces the risk of being misinformed and helps him/her get familiar with new semantics.

Insert a mapping table to resolve many : many cardinality

many:many cardinality between entities often happens in real design process when the data model is considered to be extended. (i.e. since there is an in a workspace in the real situation, occupant could be an attribute of workspace. But in the case a workspace has many occupants, the data model where 1:1 cardinality between workspace and occupant is not able to carry out 1:many cardinality.)

To solve many:many cardinality, plugging-in a mapping table between two entities is a proper resolution. Attempting to make the business logic simpler to abstract down many:many cardinality to 1:m cardinality might not be able to describe the expected query result.

Use sequential and automatically generated primary key

Naming the primary key in accordance with its naming convention might not be efficient as the number of data rows increase. That is because if the number of data rows exceeds the digit number of primary keys, there is no way but to change format of primary key. Instead, putting a meaningless sequence as a primary key would give the user much flexibility. Oracle provides a sequence function such that whenever a creation data occurs, its index is one more than the last index automatically.

Be aware of MVC concept and keep up MVC scheme

Since the finest granularity of the data model is an entity and their relations are also defined in the entity, this data schema would correspond to MVC software architecture. In other words, entities could be converted into 'Model'. 'Controller' specifies the attributes of entities and create/modify/delete each datum in the entity. Since a compartment of data model matches that of software architecture, this paradigm makes the database deployment much easier.

FUTURE WORK

In addition to fundamental functions of generic database systems, OLTP and DSS, distinct features of EnviroDB as a tool of POE, are demanded. The originality will be found in integration with GIS and real-time data acquisition.

Integration with GIS

To drill down to a specific workspace or occupant from a project is time-consuming and tedious when it has a tree hierarchy. Moreover the user does not have enough information about which work space or occupant has to be selected. For example, suppose that the user wants to see occupants' thermal satisfaction in west perimeter zone. If the database system only provides attribute level information (for example, the zone type is a 'perimeter' and the orientation type is a 'west'), not geographical information, the user has to search them from the top (project level) to the down (work space level) one by one. It is very unintuitive and inefficient because geographical information enables the user to select what spaces the user wants with almost no effort.

The query result and data trends/tendency (such as thermal profile and satisfaction intensity gradient) need to be represented geographically. As this enables the user to set the same scaled conditions in terms of building geography, the POE process will become faster and more accurate by cognitive and comprehensive comparison.

Real-time data acquisition

The real-time results are extremely significant to the POE process, since they enable the investigator to identify anomalies and trends in the observed facility, and focus on the "problematic" areas by performing further in-depth analysis or gathering more specific data [1].

Currently the measurement data / stressor data are automated through EnviroBot and EnviroSoft respectively. These data are more likely acquired first in real time. Questionnaires will be web-deployed or are to be done using mobile devices.

The proposed hardware architecture would be a server/client environment. There will be a server receiving the live data from mobile devices (currently EnviroBot and EnviroSoft). There will be a client module in each agent, which communicates with the server and transfers the data. Once the server judges that enough data are collected then it sends the data to the central database system, EnviroDB. The investigator is able to refer to current data pattern almost in real-time and make use of them to get a prompt insight.

CONCLUSION

By inherently different processes and criteria of POE, various tools, techniques and devices in data acquisition, interpretation and analysis have been developed for NEAT. EnviroDB is an integrated tool that accommodates all data/procedures of the project and offers a value to the user by means of proprietary design and functionality.

As fundamental functions of OLTP, the requirements for EnviroDB are data storage, capability to contain various types of input data/flexible data format and automated data acquisition. It also needs to suit multiple conditioned queries initiated by the user as well as enable the user to do an easy multi-factorial analysis. The analyzed data trend has to be graphically presented with the ability to compare other trends, if any. The last, a statistic summary should be reported to the analyzer.

By such requirements, the relational data model (RDM) is proposed for EnviroDB. RDM will give the user an effective and convenient resolution for the project. Additionally, some useful hands-on guidance is recommended for an ease of data model design and deployment.

After the fulfillments of the basic requirements, enhanced features such as integration with GIS and real-time data acquisition will be realized in the next steps.

REFERENCES

[1] Aziz, A; et al. 2005. EnviroSoft: A tool for POE. In preparation

[2] Aziz, A; et al. 2005. EnviroBot: Indoor environment instrument tool. In preparation

[3] CBPD, Carnegie Mellon University. 2005. Collaboration questionnaire for NEAT. CMU, USA.

[4] Software Engineering Institute, Carnegie Mellon University. 2005. Three tier software architectures <http://www.sei.cmu.edu/str/descriptions/threetier.html #34492>

[5] Wikipedia. 2005. Model-viewer-controller <http://en.wikipedia.org/wiki/MVC>

[6] Silberschatz, A.; Korth, H.; Sudarshan, S. 2002. Database system concepts 4th edition. McGraw-Hill Higher Education, 2002.